A Computational-Grid Based System for Continental Drainage Network Extraction Using SRTM Digital Elevation Models

David W. Curkendall, Eric J. Fielding, Tsan-Huei (Frank) Cheng, Josef M. Pohl
Jet Propulsion Laboratory, California Institute of Technology
{dwc, Eric.J.Fielding, Tsan-Huei.Cheng}@jpl.nasa.gov, jpohl@uwyo.edu

Abstract

We describe a new effort for the computation of elevation derivatives using the Shuttle Radar Topography Mission (SRTM) results. Jet Propulsion Laboratory's (JPL) SRTM has produced a near global database of highly accurate elevation data. The scope of this database enables computing precise stream drainage maps and other derivatives on Continental scales. We describe a computing architecture for this computationally very complex task based on NASA's Information Power Grid (IPG), a distributed high performance computing network based on the GLOBUS infrastructure. The SRTM data characteristics and unique problems they present are discussed. A new algorithm for organizing the conventional extraction algorithms [1] into a cooperating parallel grid is presented as an essential component to adapt to the IPG computing structure. Preliminary results are presented for a Southern California test area, established for comparing SRTM and its results against those produced using the USGS National Elevation Data (NED) model.

1. Introduction

In 2000 Jet Propulsion Laboratory's (JPL) Shuttle Radar Topography Mission (SRTM) flew for ten days, mapping the world for topographic elevation underneath the entire footprint of the Shuttle's path (–54º to +60º in latitude) [2]. This new database gives a grid posting every 1 arcsecond (~30m) with absolute elevation accuracy of ~ 5 meters. This grid spacing is available for the U.S.; most of the remaining world is restricted by the National Imagery and Mapping Agency (NIMA) to release at 3" (~90m) postings. Even the global release at 90m is far more accurate than large portions of the world have previously been known. The previous near-global standard for elevation models has been GTOPO30, a 30" elevation posting model of variable accuracy [3].

The focus of this study is the creation of continental–scope 'drainage extraction' maps and their closely related derivatives. Drainage extraction is the science of computing the flows of water over a given landscape – i.e., if water is poured uniformly over the land, where does it go and to what ultimate sink does it empty? The extraction of hydrologic information from digital elevation models has a long history and a well-established scientific utility. The U. S. Geological Survey (USGS) has extracted hydrologic information including the drainage networks from the global GTOPO30 dataset, producing HYDRO1k at a 1-km grid spacing for the continents [4]. They have more recently constructed higher resolution products including stream extractions from the National Elevation Database (NED) [5], called the Elevation Derivatives for National Applications (EDNA) at 30-m grid spacing [6].

The unprecedented SRTM database combines global scope and high accuracy, so it offers many opportunities to extend current hydrologic studies to nearly the entire world. These opportunities are linked to challenges stemming from the size and properties of the SRTM dataset. The near-global 1" data contains approximately 135 billion measurements and even its 3" derivative contains 15 billion, formidable challenges for extraction processing. Also, there are many small holes or 'voids' in the data arising from areas of low radar backscatter and even topographic shadowing. Finally, while the data is of high absolute accuracy and contains almost no systematic errors, the data noise is on the order of 3-5m – in areas of low slope there are many apparent sinks or local minima in the data, anathema to any stream extraction process.

In order to meet these challenges, we are implementing a methodology and a processing architecture to calculate expeditiously continental–scope drainage solutions. It will permit evolution of these solutions as the voids are slowly patched and the overall Digital Elevation Model (DEM) coverage is extended to the poles. Some of the features of this architecture are:

- Computational Grids will host the SRTM data and provide a networking and processing pipeline for obtaining the computational results.
- A parallel procedure has been developed to link together existing single threaded extraction code results into the continental–scope solutions.

Proceedings of the 2003 International Conference on Parallel Processing Workshops (ICPPW'03)
1530-2016/03 $ 17.00 © 2003 IEEE

• Results will be published on Open GIS Consortium (OGC) compatible servers and will deliver both data and image layers (Web Mapping Server (WMS) and Web Feature Server (WFS)).
• Local knowledge changes of the DEM can be easily re-submitted for recalculation, but on a local basis. The same parallel procedures that link local results together can also be used to propagate the effects of local changes to the entire drainage system.

Figure 1 The SRTM instrument employed two antennas 60m apart to measure the topography

2. SRTM project background

In February 2000 the SRTM flew for eleven days on the orbiter Endeavor [2]. A joint project of NASA and NIMA, its objective was to use synthetic aperture radar interferometry to collect data sufficient to generate digital elevation maps of the 80% of the Earth's landmass that lies between ± 60° latitude. NIMA's specification had called for vertical errors smaller than 16 meters absolute (relative to the center of the Earth) at the 90% level, with data points spaced every 1 second of arc in latitude and longitude (approximately 30 meters at the equator.) Performance evaluations by NIMA, the USGS and the SRTM project have shown these errors to be much smaller, with the most reliable estimates being approximately 5 meters.

2.1 IPG architecture for elevation model processing

"Grid Computing" is simply the logical extension of "Distributed Computing" but for high-end machines and networks - distributed supercomputing if you will. More formally: Computational Grids are persistent networked environments that integrate geographically distributed supercomputers, large databases, and high end instruments. These resources are managed by diverse organizations in widespread locations, and shared by researchers from many different institutions. There are several discreet instances of Grids in the U. S. including the Teragrid, an NSF funded partnership between several supercomputing centers, and NASA Ames Research Center (ARC): The Information Power Grid program (IPG).

These Grids have become sufficiently mature to do real science, perhaps in ways not possible without the one sign up, shared facility approach that is at the heart of the Grid access facilities. Many elements of the National Virtual Observatory [7] are being implemented on Grid Infrastructure and we plan a similar 'persistent' framework to perform elevation model processing in general and stream extraction in particular. We can build simple interface mechanisms that allows even the small and thinly connected user to reach the data collections, bring them to computational centers for large scale computing and return the results. With this vision fully implemented the data and computational cycles live on the Grid, with the software. The individual researcher would need only a networked workstation and a research plan to do significant computational science.

Figure 2 illustrates the computational network and archiving we are using throughout this project. A Grid Portal node resides at JPL, initially for submitting jobs to the IPG, but more importantly for hosting the source data, the results as they are obtained, and the serving of these results in this case along with the original SRTM elevation model as desired. For the flexible hosting of these data and allied data sets, we have constructed a 40 Terabyte data store by organizing some 160 IDE drives under the transparent management of ten Linux PC's. This system, termed RASCHAL [8], is fully RAIDed and appears as an attached disk system for up to four hosts simultaneously. Two of these are shown in the figure: Tejat, the IPG portal, and Jaba which in this case scalably serves these data in response to internet requests. RASCHAL also faces the Grid as shown and participates with Tejat in submitting and managing the computations and data flow.

JPL is connected to ARC via NASA's Research and Educational Network; at ARC, several machines as shown are managed by GLOBUS and present themselves uniformly to credentialed user's of the IPG.

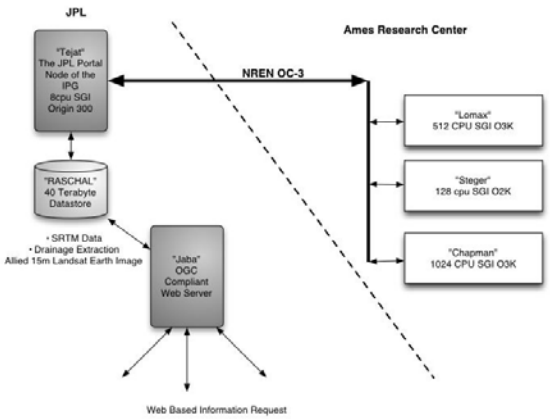

Figure 2 IPG Architecture for Elevation Model Processing and Results Publication

2.2 Elevation model derivatives and the stream extraction procedure

The simplest elevation derivative is a slope map, the first derivative of the DEM. Despite its simplicity, the local slope is one of the most important characteristics of the Earth's surface.

The local slopes can also be followed downhill to determine where water falling on the surface will go (assuming that it doesn't sink into the ground). This is the process of drainage extraction to calculate a stream network that routes the water over the ground surface to the ocean or other final destination [1]. If the DEM were a perfect representation of the land surface, then the drainage extraction would be as simple as described above. Unfortunately, most DEMs have elevation errors that cause local "sinks" or places where the streams run into a false local minimum. There also can be real closed drainage basins, such as Death Valley, California. The drainage extraction process should remove the false "sinks" but keep the real ones. The procedures described by Jenson and Domingue [1] are the most commonly used. These are discussed further below.

The SRTM data we use in this project has a greater level of "noise" than most other DEMs high-spatial-frequency elevation errors that cause a much larger number of false sinks especially at the full resolution of SRTM.

The SRTM DEMs that are going to be or are already available in 2003 have an additional defect for drainage extraction: data gaps or grid cells with no elevation values. A comprehensive solution to the problem of SRTM data voids is outside the scope of this project. We are fortunate in the United States to have complete DEM coverage of the country by the National Elevation Database (NED) [5], which has a 1-arcsecond-grid posting, exactly the same as SRTM. We will use the NED to fill in the gaps in the SRTM data for the USA and other data elsewhere (with the global GTOPO30 as the last resort). More advanced methods might involve feathering to avoid sharp discontinuities at the boundaries [9].

2.3 Algorithms

We will be using several algorithms in the process of going from the raw SRTM data to the finished drainage products. Care is taken to make the initial algorithms modular, allowing for their ease of replacement as more is learned about the processing of this unique data type. For example, several pre-filters might be tried to remove the false sinks while retaining meaningful terrain dips caused by modest streambeds.

Most of the following algorithm description and overall processing scheme is based on the paper by Jenson and Domingue [1], on the sequence recommended in the Arc/Info GRID reference manual (on-line documentation), and especially on the procedures developed at the USGS Eros Data Center for the processing in the Elevation Derivatives for National Applications (EDNA) project [6]. The EDNA project has derived a drainage network for all of the USA from the NED at the full 30-m grid resolution, approximately 60 GB of DEMs.

Beyond the techniques used by the USGS, our focus will be to:

- Modify the data and the data conditioning steps to account for SRTM-specific characteristics – many small voids and noise induced false local minima.
- Tailor the applications to the IPG, using the facilities of Globus to schedule computations, route data flow, and capture results.
- Add a parallel drainage 'linkage' of the individual blocks of terrain and use the resulting block–linking matrices as the basis for establishing an efficient drainage update process.

Local slopes (and higher order derivatives such as curvature) can be calculated over various length scales by using different "window" or kernel sizes in the computation.

The results of the local slope calculation are not used directly in the following steps, so this can be done completely independently.

The simplest method of reducing the elevation noise errors is to smooth the data with a "boxcar" filter. This is a spatial convolution filter that just takes the average elevation in a window. For larger windows, a filter weighting can be applied, such as a negative power of the distance from the given point. The output of the previous step of gap filling and the elevation error reduction will be a version of the SRTM DEM that can be processed the same as the original by following steps, so

these steps are very easy to modify by plugging in a different procedures.

We use the standard flow direction algorithm of Jenson and Domingue [1], which determines the direction that water will flow from a given cell. This is now usually called the "D8" algorithm because it determines which of the 8 neighbors of a given DEM cell is most "downhill."

If the cell is lower than all of its neighbors, then it is a "sink" or local minimum and gets a "sink" flag.

Filling the false sinks is one of the most complex steps in the drainage extraction process. It normally requires a substantial amount of iteration, because removing one sink may create another one. We skip the specifics here.

Once the flow direction for all of the cells in the DEM are resolved, the flow accumulation is easy to calculate. The cells into which no neighbors flow are set to zero flow accumulation. Then we take the accumulated flow, plus one for the current cell and pass it along into the next "downhill" cell. The result is a count of the number of cells that drain into a given cell. This can easily be converted to upstream area by multiplying by the cell area.

We use a simple threshold on the flow accumulation. This threshold controls the drainage density or length of stream channels per square kilometer. The result is a raster with "1" where the flow accumulation is above the threshold and "0" elsewhere. This is a raster version of the drainage network.

We use the scheme described by Strahler [7] where the first stream (where the flow accumulation first exceeds the threshold) is order 1. Then when two order-1 streams join, the new larger stream is order 2. The intersection of two order-2 streams form an order-3 stream. Note that any number of order-*n* streams can join a stream of higher order (*n*+1 or greater) without changing the order designation.

3. Extending the algorithms to the fully parallel domain.

It is clear that the process of determining the water flow upon the land is, in some ways, almost a perfectly data–parallel process. It rains everywhere and the produce of that rain starts independently at each pixel of Earth and runs downhill. But there undeniably are very sequential aspects of this process as well.

In what follows, we posit a parallel algorithm that begins with an equal-area domain decomposition. We will assume that the stream extraction over each block will be performed with the single threaded algorithm already described above and we concentrate here on how the various pieces can be linked together to create a complete solution.

The motivations for considering such an approach include the removal of the requirement to first have the sub-basins delineated and the much more convenient rectangular or square shape of the blocks. The setup for truly Continental runs will be straightforward, load balancing should be nearly automatic, and it will prove easier to rerun as the SRTM database evolves – the voids are filled and new datasets extending to the poles are gathered.

3.1 Algorithm introduction and basic notions

In what follows, we will focus on the three main global issues:

- Calculate the total stream flow within the network as a sum of the contribution from each of the individual blocks.
- Provide for the updating of the stream order as it would be calculated globally.
- Provide a procedure that will unify the calculation of the individual *drainage basins* across the various partitions calculated independently.

Assume we have an area, decomposed by node into regular portions or blocks of the terrain as depicted in Figure 3. In general, there will be edge nodes, shown cross-hatched, that denote the end of the database. In the case shown, the west coast partially fills the left most column of blocks, whereas the remainder of the perimeter is the edge of the database being considered here.

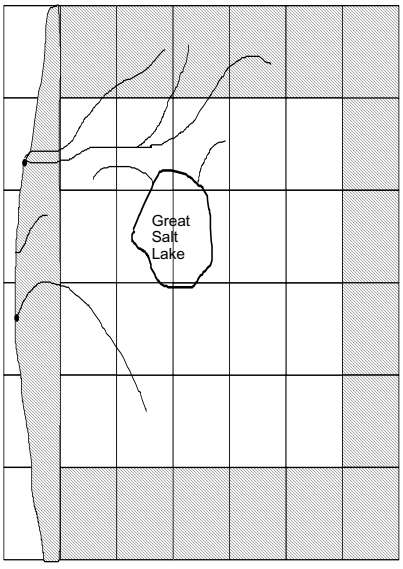

Figure 3 Regular Grid Decomposition With Ocean and Great Salt Lake Sinks

Figure 4 shows a little more detail of two contiguous and communicating blocks. These blocks are themselves

composed of equal area atomic pieces of land, called cells, and the border of the areas will be composed of 'edge' cells. Normally, the edge cells of a given block will be considered a one dimensional array of cells, with the position in the array determining which side of the local block it borders – N, E, S, or W.

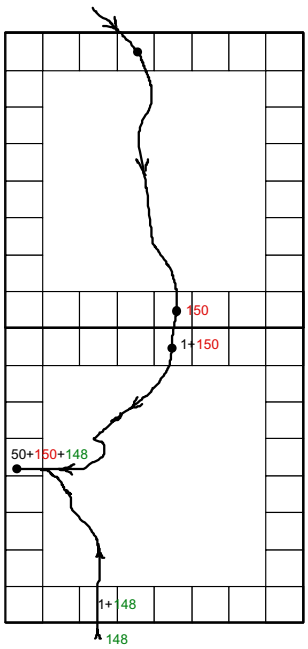

Figure 4 Two Adjacent Node Areas With Edge Cells Emphasized.

Most edge cells will be either an input cell – i.e., it will receive drainage from the neighboring block – or it will be an output cell – it will be a contributor to its adjacent area. But there are exceptions. Consider Figure 5. Illustrated here is the case where a single edge cell is receiving flow from its three closest neighbors in the adjacent block (this is a limiting condition except at a corner). In this case, the two cells on either side may not communicate with the adjacent area and are themselves in effect 'headwater', or ridge cells.

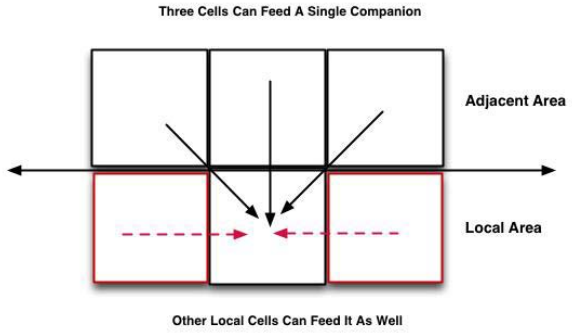

Figure 5 Input Or Output And With Whom

Up to five cells in the adjoining blocks at a corner (and all but one of its surrounding cells in the local area!) can drain to a single border cell. This in turn means that some of the local cells, don't have a companion in the adjacent block. To compute this border activity, each block will have to be augmented by the companion border cells in the adjacent blocks. Each node gets one row or column of the adjacent block to determine the category and neighbor link for each cell. Those computations are duplicated in each of the nodes but give commensurate results – implicit coordination if you will. That is, if a local border cell computes itself to be an input cell, it follows that the adjoining nodes would judge its companion to be an output cell and everything stays coherent.

An input border cell receives its water (or cell count) from an output cell in the adjoining area. This has proved confusing, so in general, we will speak of input and output cells (meaning *local* I/O cells) and their *companions* (meaning cells in the adjacent areas of opposite gender to those they communicate with).

We have the internal stream network extracted for each block, and that this gives us the ability to relate the input and output cells to each other. That is, we can construct the following records for each output cell:

$$E_o[i] = \{O, n_w, n_i, S\#\}$$

where: O = the (provisional stream order)

n_w= # of cells, draining to this point (at first just from this local area)

$n_{i=}$ # of input cells pointing to this output cell

S# = serial number of stream. To be assigned later.

Then for each E_o, there is an n_i dimensional vector containing the list of edge cells pointing to this outlet. This mapping is obtained by starting at each input cell and following the flow field computed locally as described above. When an output cell is reached, the input cell being followed downstream gets filled in this data structure and n_i is incremented.

We also denote a similar record for all of the Input cells

$$E_i[j] = \{O, n_w, E_o(comp), S\#\}$$

Here O,n_w will initially be zero but will later be filled in as the stream order, # of cells drained as received from its companion cell in Phase III of the processing as described below, respectively.

E_o(comp) is the unique identifier of the companion cell(s) in the adjacent area that empty to this cell – and we have just illustrated that the dimension of E_o(comp) can be as high as five. And S# is the same as is used in its corresponding output cell and will be assigned in Phase II of the processing.

To summarize the rules for the input and output border cells of a given block:

- Each input cell will associate with one and only one output cell. (streams merge, they don't fork)
- Many input cells can map to the same output cell. (streams merge)

• Output cells need not have any sources for them outside the local area. That is, some output cells may have no linked input cells. (the headwaters for every stream has to be somewhere)
• Some of the edge cells will not be assigned as either input or output – see the example in Figure 5.

The two above data structures are designed to disclose what's going on at the perimeters and can be used to transfer information up and down the drainage basins without the interior of the block.

3.2 Phase I processing

The processing internal to each data segment proceeds using the Jenson and Domingue [1] algorithms as outlined above. The mapping of input to output cells then proceeds in straightforward fashion and the above data structures are filled out.

3.3 Phase II processing - Up the Columbia

The cells on the boundaries of oceans need to be handled as in edge cells except we recognize that these are the ultimate *sink* cells. Each one must be inspected to see if they map to any of the block's input cells. If they do, a unique identifier must be assigned to this far reaching stream (river).

For each such ultimate sink that sources outside the local data partition:
• Assign a stream id
• Transfer this to each of the list of input cells (remember that there are n_i of these) that are linked to this S#; fill in that field of each such Ei record.
• Each of the so designated Ei records will in turn transmit this id to its companion cells with normal message passing procedures.
• The companion (output) cells alerted in this way will again forward the stream id's to its own input cells, filling in the stream id's as it proceeds.
• Continue with this activity until each id has reached nodes that have no input cell corresponding to the output with that id. That is, we have reached all of the headwaters of that stream.

For flow control and deadlock prevention, it is necessary for each node to forward these messages upon each receipt rather than wait until all have been received. Meandering rivers can go in and out of local areas; waiting can produce deadlock. When all of this activity for all ultimate sinks has been fully transmitted and the machine is once again quiet, it is time for Phase III processing.

3.4 Phase III processing - All together now, pour!

Our hard work can now be rewarded. Each block node will inspect each of its output cells. Output cells without a linked local input cell can transfer its accumulated drainage, n_w, to the companion in the adjacent area. A cpu node receiving this new information, updates its appropriate local input records and then uses those updated records to update its corresponding output cell. Recall that each of these output cells knows how many local input cells it needs to hear from, n_i, until it has received all of the accumulated water its going to get. It waits until it has heard from everyone. Then a message is sent downstream to the next area and so on until the ocean is once again reached.

When all of the streams have been fully processed, each node can update its internal flow accumulation values. That is, whenever a given block is complete as to these border issues, the water being received by the input cells can be distributed in accordance with the network already established by the Phase I processing by simple superposition. Keep these activities going until all of the ultimate sinks have been updated. The global stream network is now extracted.

3.5 Computing drainage areas

Jenson and Domingue [1] describe a procedure for calculating the drainage basin for a given system or point on a river. A 'seed' is placed on the network, usually at some prominent point like the confluence of two streams or perhaps where a monitoring station or dam is to be placed on the real terrain. The flow field is used to identify all cells flowing immediately to that point. These points are colored the same as the seed point. All the cells draining to each of these new cells are in turn identified as being members of the same drainage. This procedure continues until a ridge, or divide, is reached by all threads and the basin is defined.

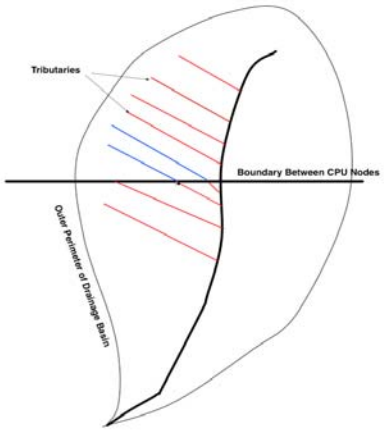

Figure 6 Calculating Drainage Basins in Parallel

Now consider the simple drainage system illustrated in Figure 6. We draw it as an area being extracted by two separate cpus with the data partitioned as shown. If this data were to be processed all on one cpu, a properly defined single drainage area would emerge as shown. But note that in the two node case, if the top node were to do this calculation independently, it would have no way of knowing that the blue tributaries were part of the main system. Thus the basin extraction procedures can only be executed post Phase II processing. Then the tributaries that cross the node boundaries will have been identified as part of the main system. The basin algorithm needs to be modified only in that each output edge cell becomes a seed whose membership is resolved by the Phase II process.

Note also, if sub-basins are to be separately identified, an additional technique needs to be used that hierarchically elaborates the stream serial numbers as bifurcations take place at stream junctions. For example, if the Columbia River is a stream of order N, it needs to be separated into Columbia.1 and Columbia.2 when the two N-1 streams converge to promote the resulting stream to Nth order. This is easily implemented.

3.6 Additional considerations

What the above algorithm does is to take a fundamentally sequential process – start at Lemhi pass and work your way to New Orleans – and break it regularly into chunks for mostly chunk–local calculations. These are all done in parallel on the Grid. It should be clear merging the resulting components together has many sequential aspects. However, if we have several river systems mapped, the overall algorithm is constructed so that many activities can be executed in parallel even in phase II and III processing. But now the order of parallelism is roughly the number of river systems instead of the number of cpus as the case in Phase I processing.

If these remaining phases are even largely sequential, a lot has been gained. The number of input and output edge cells will be a maximum of 4N for an N x N block size, much smaller than the total number of cells (N^2) in the block. This means that the Phase II and III computations, except for the final updating of the interior cells, can be done with fewer nodes, perhaps in a single large machine, to reduce the cost of the large amount of communication between blocks.

What if an entire interconnected drainage system (or maybe better posed as 'entire Continent') can't fit in even the largest Grid computer at once? Can successive runs that take this approach be glued together? The answer is yes.

Referring to Figure 7, suppose the North American Continent needs to be broken into two runs to be accommodated. If each is completed through what we have called Phase I processing and all of the 'edge' data structures are saved from both runs, they can all be loaded for the Phase II and III processing without all of interior data. Then they are treated exactly as already described.

Note also that after the edges have all been updated to the more global result, the detailed processing interior to those edges can now take place without further coordination. That is, post processing is still completely parallel.

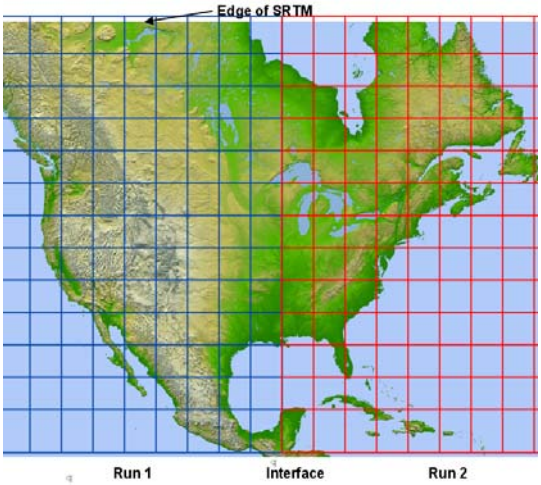

Figure 7 North America in Two Passes

4. Preliminary Results

4.1 Target final products

The following products are being produced, first for the portion of the North American continent covered by SRTM data. Extensions seek to consider each continent in turn until a global set of results is in place.

1) Slope map grids, with magnitude and direction of slope.
2) Curvature map grids, with along-slope and across-slope curvature (or similar measures).
3) Hydrologically conditioned DEM, after false sinks have been removed.
4) Flow accumulation map grid.
5) Flow direction map grid
6) Stream network map grid
7) Ordered stream network map grid
8) Ordered stream network vector topology.

The above products will be made available via the OpenGIS standards, initially using the WMS protocol to provide images derived from the data products.

4.2 Preliminary Products Produced to Date

Comparisons have been made between the SRTM and NED topography for a test area with a variety of topographic types roughly 300 x 180 km, in southern California (Figure 8). Part of this area is covered by a high-resolution digital elevation model made by the NASA/JPL TopSAR system, an airborne Interferometric SAR, with 10-m grid spacing. Prototype non-parallel versions of the drainage network extraction scheme and surface slope angle calculations were run on all three of the databases.

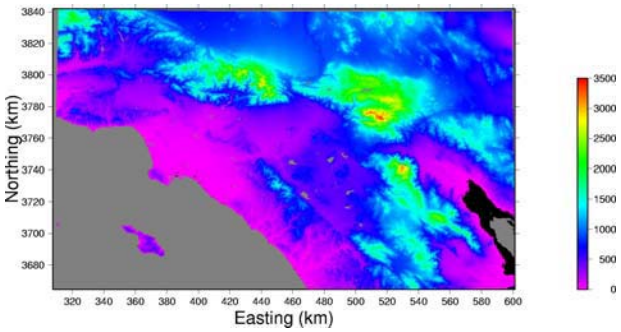

Figure 8 Shuttle Radar Topography Mission data for southern California test site. Gray areas are ocean or voids (mostly lakes and Salton Sea at lower right). The black area, adjacent to the Salton Sea, is below sea level. Coordinates are in UTM zone 11.

Preliminary results from the SRTM-NED-TopSAR comparison [10] are shown in Figures 9-15. The slope calculations from the three topographic datasets produced similar results (Figures 9, 10, 11-13) but had some systematic differences due to variations in the accuracy and resolution of the datasets. Slopes were calculated by fitting a quadratic surface to windows of 9 x 9 pixels. In Figures 9 and10, the slopes from SRTM and NED are indistinguishable at the regional scale, except for the voids in the SRTM.

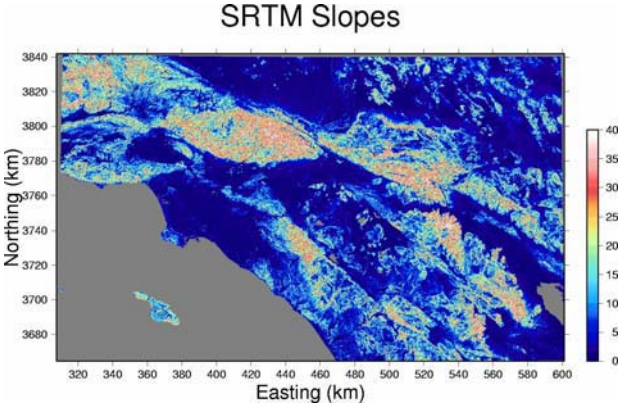

Figure 9 Hill slope angle magnitudes (degrees) calculated from SRTM. Gray areas are ocean or voids.

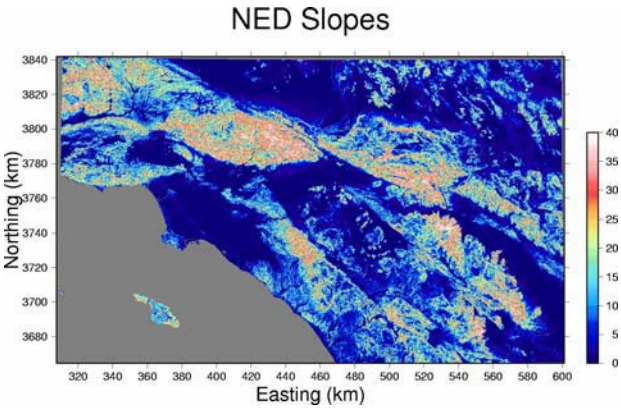

Figure 10 Hill slope angle magnitudes (degrees) calculated from NED. Gray areas are ocean.

In Figure 11, the slope magnitudes are compared for a small area in the eastern San Gabriel Mountains that is covered by all three databases. This is one of the most rugged areas in southern California, with very steep slopes. Note that the color scale of Figure 11-13 is different from that in Figure 9-10 to avoid saturating the steeper slopes. At this map scale, there is a subtle difference between the SRTM and NED slopes. The NED slopes are generally a little steeper. This is probably due to the smoothing that was applied in the processing of the SRTM data. The smoothing rounds the topography and tends to reduce the steepest slopes. This difference in resolution was measured a different way by Smith and Sandwell [11].

The TopSAR DEM, with its finer spatial resolution of 10 meters, includes substantially steeper slopes (Figure 13). The increase in slope angles with finer grid spacing is a natural result of measuring slopes over different ground distances (e.g., Zhang and Montgomery [12]). In addition, the TopSAR may have a greater noise level than the other

datasets. This is visible in the flat areas where the true slopes are probably shallow but TopSAR has some "speckle" in the slope image (Figure 13)

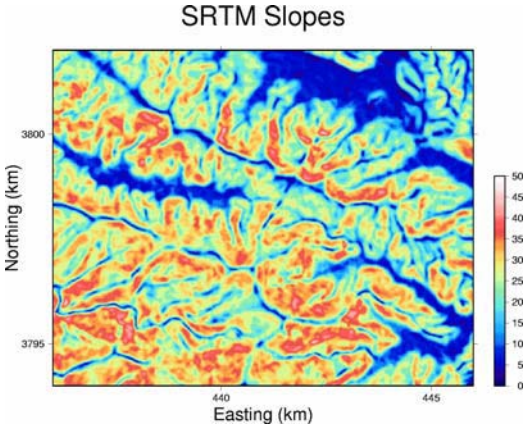

Figure 11 Slope magnitudes (degrees) for rugged San Gabriel Mountain peaks from SRTM.

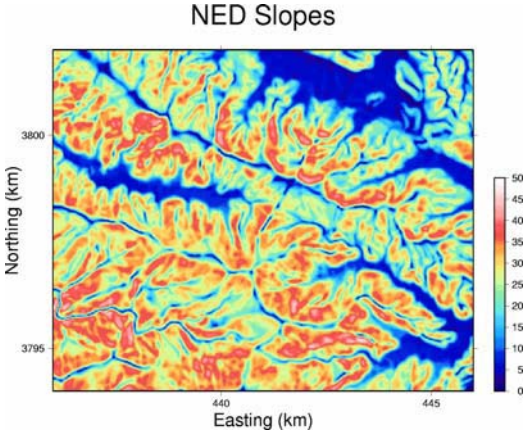

Figure 12 Slope magnitudes (degrees) for San Gabriel Mountain peaks from NED.

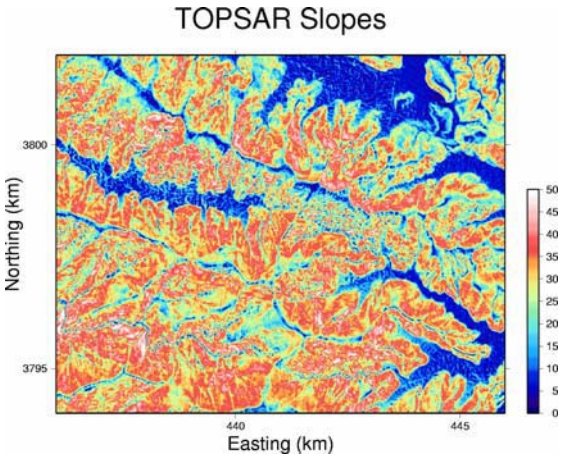

Figure 13 Slope magnitudes (degrees) for San Gabriel Mountain peaks from TopSAR.

Another way to look at the slopes is to plot the slope histograms, shown in Figure 14. The entire area of the TopSAR dataset (38 x 100 km) including the San Gabriel Mountains and part of the area to the south was used for these histograms. As described above, the TopSAR has the steepest slopes due to it finer grid spacing and higher resolution. The NED also has a slightly higher amount of the steeper slopes in the 30–40° range. The NED also has more very shallow slopes (0–1°) because it is very smooth in the flat areas, while SRTM and TopSAR have some noise that causes slopes of 1–3° in the flat areas.

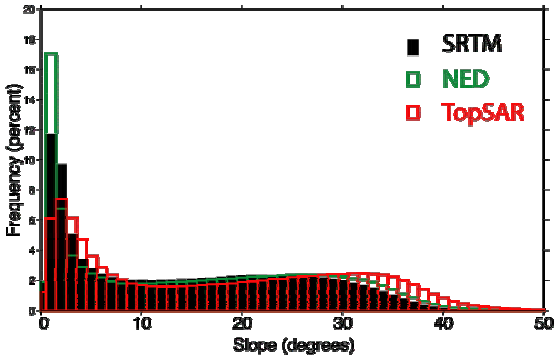

Figure 14 Slope histograms for San Gabriel Mountains and surrounding areas from three datasets.

Drainage networks were also extracted from the SRTM, NED and TopSAR topography, using the Jenson and Domingue [1] algorithm with a mixture of Fortran code and the ArcGIS software. The drainage extracted from the SRTM data (with the voids filled by NED) is shown in Figure 15, with the Strahler–orders [7] shown by different colors. Only rivers with Strahler orders >= 4 are shown. The order 4 rivers are dark blue, order 5 is cyan, order 6 magenta, and order 7 is yellow. The ocean is filled with an aqua color and the Salton Sea is filled with medium blue. The drainage networks are nearly identical in areas of mountains and hills, but show some differences in flat areas where the elevation changes are small compared with elevation errors.

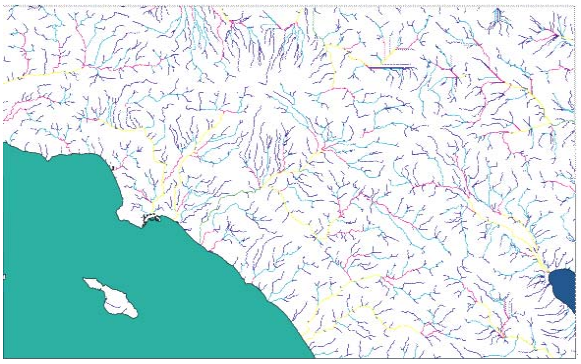

Figure 15 Drainage network extracted from SRTM data for southern California test site. Colors show the Strahler order of the rivers (see text).

5. Conclusion

We have planned a significant set of activities to exploit the SRTM dataset in order to provide a set of derived continental scale data products of scientific and practical use in the areas of hydrology, land cover/land use, ecology, biodiversity and disaster management. The key enablers for such an undertaking are the computational resources of the IPG and the algorithms presented here for fitting classical stream extraction procedures to large, parallel machines. SRTM data characteristics present unique challenges but the preliminary results recapped here give assurance that they can be met and high quality results can be obtained.

6. Acknowledgements

The research described in this paper was carried out at the Jet Propulsion Laboratory, California, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

7. References

[1] S. K. Jenson and J. O. Domingue, "Extracting topographic structure from digital elevation data for geographic information system analysis," *Photogrammetric Engineering and Remote Sensing*, vol. 54, pp. 1593-1600, 1988.

[2] T. Farr and M. Kobrick, "Shuttle radar topography mission produces a wealth of data," *Eos, Transactions, American Geophysical Union*, vol. 81, pp. 583 & 585, 2000.

[3] D. B. Gesch, K. L. Verdin, and S. K. Greenlee, "New land surface digital elevation model covers the Earth," *Eos, Transactions, American Geophysical Union*, vol. 80, pp. 69-70, 1999.

[4] K. L. Verdin and S. K. Greenlee, "HYDRO1k documentation," Sioux Falls, ND, U.S. Geological Survey, EROS Data Center, http://edcdaac.usgs.gov/gtopo30/hydro/readme.html, 1998.

[5] D. B. Gesch, M. J. Oimoen, S. K. Greenlee, C. Nelson, M. Steuck, and D. Tyler, "The National Elevation Dataset," *Photogrammetric Engineering and Remote Sensing*, vol. 68, pp. 5-11, 2002.

[6] USGS, "Elevation Derivatives for National Applications," Sioux Falls, ND, U.S. Geological Survey, EROS Data Center, http://edna.usgs.gov, 2003.

[7] B. Berriman, D. Curkendall, J. C. Good, L. Husman, J. C. Jacob, J. M. Mazzarella, R. Moore, T. A. Prince, and R. E. Williams, "Architecture for access to compute intensive image mosaic and cross-identification services in the NVO," presented at SPIE Astronomical Telescopes and Instrumentation: Virtual Observatories Conference, Waikoloa, HI, 2002.

[8] L. Plesea, Pers. Comm. 2003

[9] P. Siquiera, S. Hensley, S. Shaffer, L. Hess, G. McGarragh, B. Chapman, and A. Freeman, "A continental-scale mosaic of the Amazon Basin using JERS-1 SAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, pp. 2638-2644, 2000.

[10] E. J. Fielding, "Characterization for spatial derivatives of SRTM data to investigate controls on erosion," *Geophysical Research Abstracts*, vol. 5, pp. 13744, 2003.

[11] B. Smith and D. Sandwell, "Accuracy and resolution of shuttle radar topography mission data," *Geophysical Research Letters*, vol. 30, pp. 1467, doi:10.1029/2002GL016643, 2003.

[12] W. H. Zhang and D. R. Montgomery, "Digital elevation model grid size, landscape representation, and hydrologic simulations," *Water Resources Research*, vol. 30, pp. 1019–1028, 1994.